

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA



Estructura de Computadores I

Arquitectura de los MMIOFPS

Integrantes:

Luis Castro

Valentina Yévenes

RESUMEN

Los MMOG (Massively Multiplayer Online Game), son juegos en donde pueden participar, e interactuar en un mundo virtual un gran número de jugadores del orden de cientos o miles simultáneamente conectados a través de la red. Debido a la gran cantidad de tipos de MMOG, este trabajo está centrado en el tipo MMOFPS (Massively Multiplayer Online First Person Shooter) y en el tipo de arquitectura que este utiliza.

INTRODUCCIÓN

Los MMO son videojuegos herederos de los MUD o videojuegos en línea no gráficos, nacidos en la década de los 1970. Para finales de los 90 el concepto de juegos en línea multijugador masivos había traspasado las fronteras hacia nuevos géneros como los juegos de estrategia o los juegos de acción en primera persona. Surgió la etiqueta MMOG (Massively Multiplayer Online Game) para agrupar a todos ellos con independencia de su género.

Los MMOFPS combinan los géneros del FPS y los MMO, ofreciendo un mundo persistente a un gran número de jugadores simultáneos, en el cual enfatizan la habilidad del jugador más que sus estadísticas, ya que ningún número de bonus en el juego compensara la incapacidad del jugador de apuntar y pensar tácticamente. Y es así, como llegamos al tema que nos interesa, la arquitectura de red de los MMOFPS.

¿QUÉ ARQUITECTURA SE UTILIZA EN LOS “JUEGOS ONLINE”?

En la actualidad existen diferentes arquitecturas las cuales presentan tanto ventajas como desventajas dependiendo la situación en las que se estudie (escalabilidad, complejidad de la implementación, etc.) y dependiendo del juego se debe adaptar a cada fase de él un enfoque diferente.

Entre estas arquitecturas nos encontramos con la Cliente – Servidor y Peer – to – Peer, entre ellas la propuesta más utilizada es la cliente – servidor en su mayoría por la facilidad en la que este se implementa y se controla. En este modelo, el cliente envía un mensaje

solicitando un determinado servicio a un servidor. Haciendo uso de esto el servidor, una vez procesada la petición, envía uno o varios mensajes con la respuesta. El servidor almacena los datos envidados por cada uno de los jugadores, lo que le permite a estos mismos mantener su estado actual y el de todo el mundo virtual. De la misma forma existen otras tareas que el servidor debe asumir. Debe ocuparse de las cuentas de datos de todos los jugadores, cambio de habilidades, pertenencias dentro del juego. Pero también presenta sus desventajas, ya que este modelo no es muy escalable. Solo puede haber un número limitado de servidores. Entonces, la máxima cantidad de jugadores está limitada por la cantidad de tráfico que soporten los servidores. Además debido a la alta demanda de jugadores el mundo virtual se divide en pequeñas regiones llamadas celdas, las que tienen un tamaño y asignación del tipo dinámica, lo que permite gestionar las concentraciones espontáneas de jugadores. La división del mundo virtual en un mayor número de celdas de menor densidad de jugadores permite una asignación más flexible del micro-celda entre todos los servidores del juego, habilitando un balanceo de la carga entre servidores, repartiendo los jugadores a lo largo de todos los servidores. El inconveniente de esto es que la calidad del juego disminuye y aunque esta arquitectura escala con el número de jugadores, falta flexibilidad y el servidor tiene que tener una configuración aceptable para manejar picos de carga.

Este problema no se presenta en el modelo descentralizado (Peer - to - Peer) ya que este no necesita de un servidor por lo tanto esta arquitectura es teóricamente escalable hasta un número ilimitado de servidores. Pero, a pesar de las ventajas, asegurar una persistencia de datos del juego en una red Peer-to-Peer es un desafío mayúsculo debido a que las infraestructuras distribuidas para el almacenaje de información están enfocadas a redes Peer to Peer de compartición de ficheros y rara vez cumplen con los requerimientos de seguridad que precisan las aplicaciones tipo MMOG como por ejemplo las trampas o cheats. Otro inconveniente son el ancho de banda asimétrico y el alto consumo de CPU. Finalmente nos encontramos con la arquitectura híbrida la cual está constituida tanto por el modelo Peer - to - Peer como la del modelo Cliente - Servidor que representa una

solución de compromiso entre la facilidad y la fiabilidad de control global del juego que aporta un sistema centralizado, la escalabilidad y bajo coste y sistema distribuido.

El sistema consta de dos áreas principales: el área central y el área distribuida. El área central se usa tanto para la ejecución de los juegos online en situaciones de no sobrecarga, como para la gestión de los servicios centrales del entorno, mientras que el área distribuida es explotada en situaciones de sobrecarga del sistema, momentos en los que se crea, bajo demanda, un conjunto de zonas para albergar las instancias no gestionadas por el servidor central. El modelo híbrido provee seguridad usando un servidor central que controla el acceso al estado del juego, este servidor central registra a los jugadores y monitoriza sus acciones. Cliente servidor carece de escalabilidad ilimitada, necesaria para los juegos online, por lo que P2P la aporta y este modelo híbrido se convierte en una arquitectura totalmente escalable distribuyendo el estado del juego a través de los jugadores. El servidor divide el mundo virtual en regiones, todas las actualizaciones o modificaciones que un jugador haga que afecte al entorno del jugador es enviado directamente al servidor central. Este modelo ahorra ancho de banda ya que los cambios posicionales son gestionados por el servidor regional en vez del servidor central.

PROTOSCOLOS UTILIZADOS.

UDP (User Datagram Protocol)

UDP es útil para aplicaciones que son sensibles en el tiempo por lo que la velocidad es un factor importante, en el sentido del procesamiento, y donde la pérdida de paquetes no es un problema.

Los juegos online en tiempo real entre varios jugadores requieren alta velocidad de respuesta en la interacción de los jugadores por eso suelen utilizar UDP. En estos juegos, los datos se envían por el juego a su servidor central y luego a otros jugadores ordenadores en tiempo real.

Dado que las posiciones anteriores son irrelevantes una vez que un jugador se ha movido, no importa si los datagramas se descartan. Si un datagrama nunca llega, el siguiente

datagrama se recibe lo suficientemente rápido para que haga poca diferencia, y el ligero retardo introducido puede ser compensado. Eventos como el movimiento de un personaje pueden enviarse como paquetes UDP no fiables, ya que un jugador enviará varios movimientos en muy pocos segundos, así que un movimiento "posterior" anula los anteriores y la percepción en el resto de los jugadores sobre el movimiento del primero es casi normal.

En cambio, hay eventos que afectan al estado del juego, como un impacto a otro personaje, recoger un objeto, etc., los que deben ser enviados como paquetes UDP "fiables" para tener la certeza que el resto de jugadores han recibido ese evento y la percepción del entorno de juego sea homogénea.

CONCLUSIONES

La mayoría de los juegos online utilizan la arquitectura cliente servidor, pero no en un 100%, durante el desarrollo de este informe nos fuimos dando cuenta de que también utiliza la arquitectura P2P y que ambas se complementan según las necesidades y tipos de juego. Aprendimos que a pesar de lo imaginado, los juegos online utilizan como protocolo de transporte UDP ya que necesitan que el transporte de paquetes sea rápido, a pesar de la pérdida que pueda haber.

REFERENCIAS

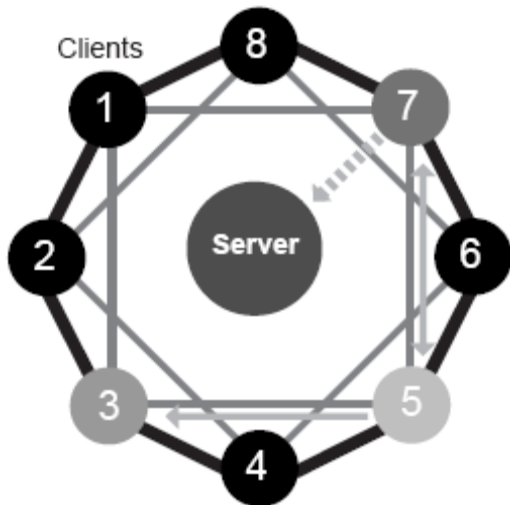
http://es.wikipedia.org/wiki/Videojuego_de_disparos_en_primera_persona

[http://neutron.ing.ucv.ve/fernandezl/Multimedia/Tareas%202007-1/Juegos%20en%20Red%20Trabajo%20Final%20\(2\).pdf](http://neutron.ing.ucv.ve/fernandezl/Multimedia/Tareas%202007-1/Juegos%20en%20Red%20Trabajo%20Final%20(2).pdf)

<http://www.tdx.cat/bitstream/handle/10803/78049/Tibv1de1.pdf?sequence=1>

<http://www.stratos-ad.com/forums/index.php?topic=246.0>

ANEXO



Modelo híbrido